

```

1  Object subclass: #Flag
2      instanceVariableNames: 'flagTeam'
3      classVariableNames: ''
4      poolDictionaries: ''
5      category: 'PLP'!
6
7      !Flag methodsFor: 'setting'
8      team: aTeam
9          flagTeam := aTeam.! !
10
11
12      !Flag methodsFor: 'accessing'
13      team
14          ^flagTeam.! !
15
16      !Flag methodsFor: 'accessing'
17      isVisible
18          ^true.! !
19
20
21  Object subclass: #Game
22      instanceVariableNames: 'terrain flags teams currentTeamIndex'
23      classVariableNames: ''
24      poolDictionaries: ''
25      category: 'PLP'!
26
27      !Game methodsFor: 'handling'
28      pass
29          currentTeamIndex := (currentTeamIndex % teams size) + 1! !
30
31
32      !Game methodsFor: 'as yet unclassified'
33      moveCharacter: aMemberIndex to: aDirection steps: steps
34          (self currentTeam members at: aMemberIndex) move: steps to: aDirection .
35          self pass! !
36
37      !Game methodsFor: 'as yet unclassified'
38      items: aClass at: positions
39          ^ (1 to: positions size) asOrderedCollection collect: [ :i |
40              | obj |
41              terrain at: (positions at: i) put: (obj := aClass new) .
42              obj
43          ]! !
44
45      !Game methodsFor: 'as yet unclassified'
46      currentWinners
47          | teamFlags flagsPerTeam maxFlags |
48
49          (teamFlags := flags select: [ :f | f team isNotNil ]) ifEmpty: [ ^{ }
50      ] .
51          maxFlags := ((flagsPerTeam := teamFlags groupedBy: [ :f | f team ])
52      collect: [ :fs | fs size ]) max .
53          ^ (teams collect: [ :t | t name ]) select: [ :tn | (flagsPerTeam at: tn
54      ifAbsent: {}) size = maxFlags ]! !
55
56      !Game methodsFor: 'as yet unclassified'
57      registerTeam: aTeamName playerLocations: aPlayerCollection
58          | players |
59          players := aPlayerCollection pairsCollect: [ :pos :unitType |
60              unitType new: terrain with: pos and: aTeamName ] .
61          teams add: (GameTeam new: aTeamName with: players) .! !
62
63      !Game methodsFor: 'initialization'
64      initializeWithX: anInteger Y: anInteger2 trapLocations: trapLocations
65      flagLocations: flagLocations
66          terrain := (Terrain dimX: anInteger dimY: anInteger2) .
67          self items: Trap at: trapLocations .
68          flags := self items: Flag at: flagLocations .
69          teams := OrderedCollection new .
70          currentTeamIndex := 1 .! !
71
72      !Game methodsFor: 'accessing'
73      currentTeam
74          ^ teams at: currentTeamIndex.! !
75
76      "....."!

```

```

75
76     Game class
77         instanceVariableNames: ''!
78
79     !Game class methodsFor: 'instance creation'
80     newX: anInteger Y: anInteger2 trapLocations: trapLocations flagLocations:
flagLocations
81         ^ self new initializeWithX: anInteger Y: anInteger2 trapLocations:
trapLocations flagLocations: flagLocations
82         ! !
83
84
85 Object subclass: #GameTeam
86     instanceVariableNames: 'name members'
87     classVariableNames: ''
88     poolDictionaries: ''
89     category: 'PLP'!
90
91
92     !GameTeam methodsFor: 'accessing'
93     name
94         ^ name! !
95
96
97     !GameTeam methodsFor: 'initialization'
98     initialize: aString with: aCollection
99         name := aString .
100         members := aCollection .
101         ! !
102
103
104     !GameTeam methodsFor: 'as yet unclassified'
105     members
106         ^ members! !
107
108     "....."!
109
110     GameTeam class
111         instanceVariableNames: ''!
112
113     !GameTeam class methodsFor: 'instance creation'
114     new: aString with: aCollection
115         ^ self new initialize: aString with: aCollection ! !
116
117
118 Object subclass: #Terrain
119     instanceVariableNames: 'matrix'
120     classVariableNames: ''
121     poolDictionaries: ''
122     category: 'PLP'!
123
124
125     !Terrain methodsFor: 'as yet unclassified'
126     dimX: anInteger dimY: anInteger2
127         matrix := Matrix rows: anInteger2 columns: anInteger.! !
128
129     !Terrain methodsFor: 'as yet unclassified'
130     elementsAround: aUnit satisfying: aBlockClosure
131         ^ (self elementsAround: aUnit) select: aBlockClosure .! !
132
133
134     !Terrain methodsFor: 'enumerating'
135     select: aBlockClosure
136         ^ (matrix asBag select: [:x | x isNotNil ]) select: aBlockClosure ! !
137
138
139     !Terrain methodsFor: 'accessing'
140     at: aPoint
141         ^ matrix at: aPoint y at: aPoint x ! !
142
143     !Terrain methodsFor: 'accessing'
144     at: aPoint put: anObject
145         matrix at: aPoint y at: aPoint x put: anObject ! !
146
147     !Terrain methodsFor: 'accessing'
148     elementsAround: aUnit
149         | points |
150         points := aUnit location eightNeighbors select: [:x | (self contains:
x) and: [(self at: x) isNotNil] ].

```

```

150         ^aUnit selectSeenFrom: (points collect: [ :p | self at: p ]) .! !
151
152         !Terrain methodsFor: 'accessing'
153         visibleAt: aPoint
154             ^ (self at: aPoint) value.! !
155
156
157         !Terrain methodsFor: 'removing'
158         remove: anObject
159             matrix replaceAll: anObject with: nil! !
160
161
162         !Terrain methodsFor: 'as yet unclassified '
163         contains: aPoint
164             ^ (aPoint x > 0) & (aPoint x <= matrix columnCount) & (aPoint y > 0) &
(aPoint y <= matrix rowCount) ! !
165
166         "....."!
167
168         Terrain class
169             instanceVariableNames: ''!
170
171         !Terrain class methodsFor: 'as yet unclassified'
172         dimX: anInteger dimY: anInteger2
173             ^ (self new) dimX: anInteger dimY: anInteger2.
174         ! !
175
176
177         Object subclass: #Trap
178             instanceVariableNames: ''
179             classVariableNames: ''
180             poolDictionaries: ''
181             category: 'PLP'!
182
183         !Trap methodsFor: 'accessing'
184         value
185             ^nil.! !
186
187         !Trap methodsFor: 'accessing'
188         isVisible
189             ^false.! !
190
191
192         Object subclass: #Unit
193             instanceVariableNames: 'isDisabled terrain location team'
194             classVariableNames: ''
195             poolDictionaries: ''
196             category: 'PLP'!
197
198
199         !Unit methodsFor: 'testing'
200         isVisible
201             ^ true.! !
202
203
204         !Unit methodsFor: 'ej4'
205         disable
206             isDisabled := true.! !
207
208         !Unit methodsFor: 'ej4'
209         interactWithAllElements
210             (terrain elementsAround: self satisfying: [:x | self target value:x ])
do: [ :item | self action value:item ]! !
211
212         !Unit methodsFor: 'ej4'
213         canMove
214             ^isDisabled not.! !
215
216         !Unit methodsFor: 'ej4'
217         canHeal
218             ^ self subclassResponsibility! !
219
220         !Unit methodsFor: 'ej4'
221         action
222             ^ self subclassResponsibility! !
223
224         !Unit methodsFor: 'ej4'
225         moveTo: aDirection

```

```

226         |content|
227         (self canMoveTo: aDirection) ifFalse: [ ^self].
228         terrain remove: self.
229         location := aDirection nextFrom: location.
230         content := terrain at: location.
231         terrain at: location put: self.
232         content ifNil: [self interactWithAllElements] ifNotNil: [ self disable
233     ].
234     ^self! !
235
236 !Unit methodsFor: 'ej4'
237 target
238     ^ self subclassResponsibility ! !
239
240 !Unit methodsFor: 'ej4'
241 canMoveTo: aDirection
242     | nextLocation nextLocationIsValid |
243     nextLocation := aDirection nextFrom: location.
244     nextLocationIsValid := [ (terrain contains: nextLocation)
245                             and: [
246                                 (terrain at: nextLocation) isNil
247                                 or: [ (terrain at: nextLocation) isVisible not ]
248                             ]
249     ].
250     ^ self canMove and: nextLocationIsValid! !
251
252 !Unit methodsFor: 'ej4'
253 healedBy: aUnit
254     aUnit canHeal ifFalse: [Exception new signal: 'Solo medicos sanos
255     pueden curar unidades'].
256     isDisabled := false . ! !
257
258 !Unit methodsFor: 'ej4'
259 move: anInteger to: aDirection
260     anInteger timesRepeat: [ self moveTo: aDirection ] ! !
261
262 !Unit methodsFor: 'ej4'
263 isDisabled
264     ^ isDisabled ! !
265
266 !Unit methodsFor: 'as yet unclassified'
267 team
268     ^ team! !
269
270 !Unit methodsFor: 'as yet unclassified'
271 selectSeenFrom: aCollection
272     ^ aCollection select: [:x | x isVisible ] .! !
273
274 !Unit methodsFor: 'as yet unclassified'
275 location
276     ^ location! !
277
278 !Unit methodsFor: 'initialization'
279 initializeIn: aTerrain with: aPoint and: aString
280     team := aString .
281     isDisabled := false .
282     terrain := aTerrain .
283     location := aPoint .
284     aTerrain at: aPoint put: self.! !
285
286 "... .."
287
288 Unit class
289     instanceVariableNames: ''
290
291 !Unit class methodsFor: 'instance creation'
292 new: aTerrain with: aPoint and: aString
293     ^ self new initializeIn: aTerrain with: aPoint and: aString! !
294
295 Unit subclass: #Engineer
296     instanceVariableNames: ''
297     classVariableNames: ''
298     poolDictionaries: ''
299     category: 'PLP'!
300
301

```

```

302
303     !Engineer methodsFor: 'ej4'
304     target
305         ^ [ :x | x isKindOf: Trap ]! !
306
307     !Engineer methodsFor: 'ej4'
308     action
309         ^ [ :x | terrain remove:x ]! !
310
311     !Engineer methodsFor: 'ej4'
312     selectSeenFrom: aCollection
313         ^ aCollection .! !
314
315     !Engineer methodsFor: 'ej4'
316     canHeal
317         ^false! !
318
319
320     Unit subclass: #Explorer
321         instanceVariableNames: ''
322         classVariableNames: ''
323         poolDictionaries: ''
324         category: 'PLP'!
325
326
327     !Explorer methodsFor: 'ej4'
328     target
329         ^ [ :x | x isKindOf: Flag ]! !
330
331     !Explorer methodsFor: 'ej4'
332     canHeal
333         ^false! !
334
335     !Explorer methodsFor: 'ej4'
336     action
337         ^ [ :x | x team: team ]! !
338
339
340     Unit subclass: #Medic
341         instanceVariableNames: ''
342         classVariableNames: ''
343         poolDictionaries: ''
344         category: 'PLP'!
345
346     !Medic methodsFor: 'ej4'
347     target
348         ^ [ :x | (x isKindOf: Unit) and: [ x isDisabled and: [ x team = team ]
349         ] ]! !
350
351     !Medic methodsFor: 'ej4'
352     canHeal
353         ^isDisabled not ! !
354
355     !Medic methodsFor: 'ej4'
356     action
357         ^ [ :x | x healedBy: self ]! !

```